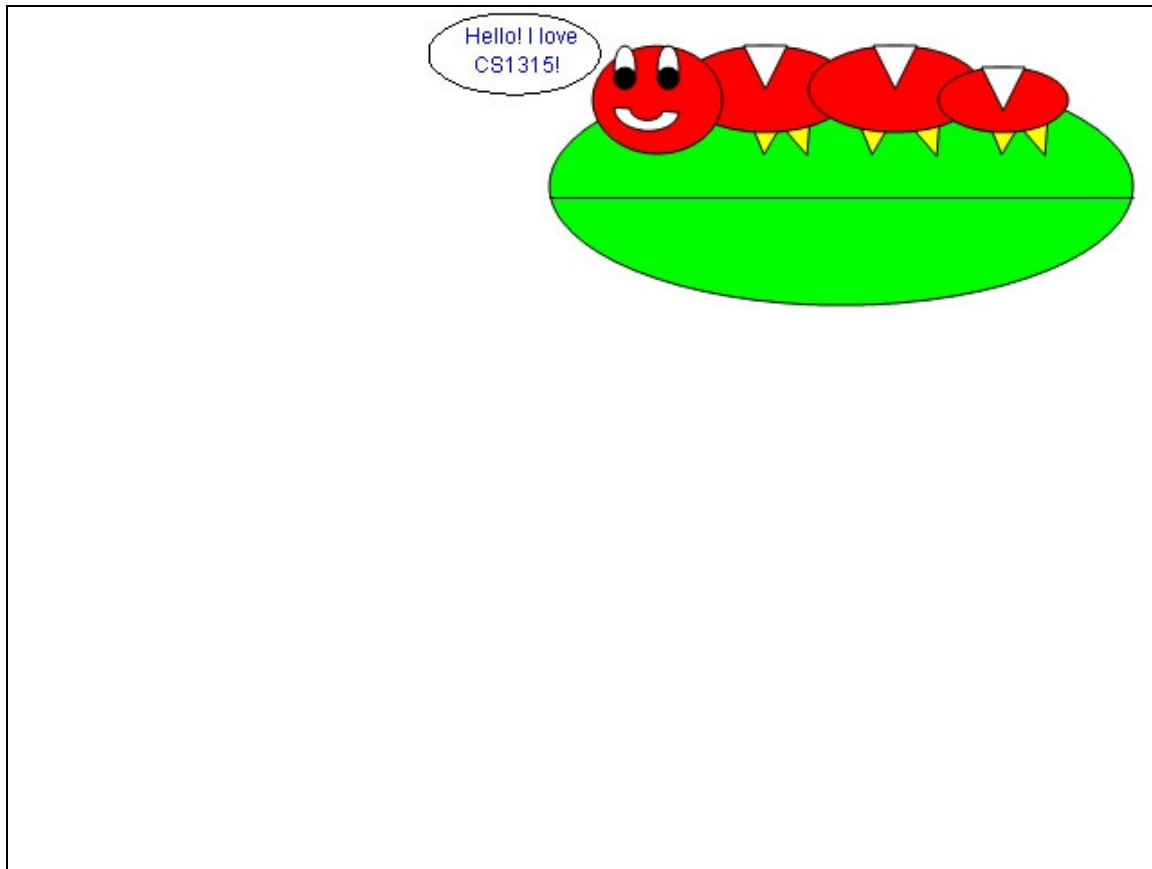


How To Make A Speech Bubble

By: Sweta Vajjhala

There can be many different applications of picture-manipulation in JES. One of the best applications is making a comic strip. In these comic strips, there are often characters that say things. To indicate that a character is speaking, a speech bubble is often used. JES can be used to make speech bubbles for these characters. This tutorial is a step-by-step tutorial that will show an example of how to make a speech bubble.

Our desired result is as follows:



Assuming that all of this should be done inside of one function, then there are three main steps required in order to obtain the above result:

- 1) Make the canvas
- 2) Make & copy the character onto the canvas
- 3) Make the speech bubble & add it to the canvas.

Making The Canvas

First, we have to define a function: let's call it `makeSpeechBubble`. Also, be sure to set your media path to the place where your character and canvas are located. We will also assume that we are making a picture from a pre-existing canvas, which has a size of 640x480 pixels. Once we add the canvas, we want the entire background to be white. So far, this is what our code looks like:

```
def makeSpeechBubble():  
    canvas = makePicture(getMediaPath("640x480.jpg"))  
    canvas.addRectFilled(white,1,1,640,480)
```

Making And Copying The Character Onto The Canvas

After making the canvas, the next step is to copy a character onto the canvas. In order to do this, first, we have to make the picture, which is similar to making the canvas.

```
caterpillar = makePicture(getMediaPath("caterpillar.jpg"))
```

The easiest way to copy the caterpillar onto the canvas is using two for loops. For each pixel (x,y), we will copy it onto the canvas at location (targetX,targetY), where the first value for targetX and targetY will start at the value where the top-left corner of the image should appear.

```
targetX = 300  
for x in range(1,getWidth(caterpillar)):  
    targetY = 20  
    for y in range(1,getHeight(caterpillar)):  
        c1 = getColor(getPixel(caterpillar,x,y))  
        setColor(getPixel(canvas,targetX,targetY),c1)  
        targetY = targetY + 1  
        targetX = targetX + 1
```

This means that we will copy the caterpillar onto the canvas, and the top-left corner of the caterpillar will appear at (300,20) on the canvas.

Making and Adding The Speech Bubble Onto The Canvas

Now that we have a character that is copied onto the canvas, we have to make a speech bubble for the character. In order to make the bubble and add it to the canvas at the same time, there is a built-in JESS function called `addOval`. This function takes in five parameters (in the following order): color, startingX, startingY, width, height. This will give you an oval

```
canvas.addOval(black,235,5,95,45)
```

Now that we have our actual speech bubble, we need to put the speech inside of it! Just like there is an `addOval` function, JES also has a built-in `addText` function. The parameters for the `addText` function are: color, startingX, startingY, "text". It is best to specify the startingX and the

startingY somewhere inside the bubble. For example, the speech bubble itself starts at (235,5), so starting the text at (235,5) would not be a good idea. Instead, starting the text at (255,22) ensures that the entire text will appear inside the speech bubble. Since I was “CS1315!” to appear in a new line, I will start that text at (260,28). This will give the appearance that “CS1315!” is centered under “Hello! I love”.

```
canvas.addText(blue,255,20,"Hello! I love")  
canvas.addText(blue,260,38,"CS1315!")
```

After all of this is done, it is important to remember to return the canvas. If you want to save the final result, you can put in a **writePictureTo** statement, which will write the canvas to a file. However, if you want it to be executed, it must be placed before the return statement.

```
writePictureTo(canvas,"C:\\Sweta\\Georgia Tech\\CS 1315 TA\\caterpillar.jpg")  
return canvas
```

When you run this program, it should be run as follows:

```
>>> setMediaPath()  
New media folder: C:\Program Files\JES\MediaSources\  
>>> show(makeSpeechBubble())
```

After running this, the final result should be the picture above!